

HORN VERSUS FULL FIRST-ORDER: COMPLEXITY DICHOTOMIES IN ALGEBRAIC CONSTRAINT SATISFACTION

MANUEL BODIRSKY, PETER JONSSON, AND TIMO VON OERTZEN

CNRS/LIX, École Polytechnique, France
e-mail address: bodirsky@lix.polytechnique.fr

Department of Computer and System Science, Linköpings Universitet, Linköping, Sweden
e-mail address: petej@ida.liu.se

Max-Planck-Institute for Human Development, Berlin, Germany
e-mail address: vonoertzen@mpib-berlin.mpg.de

ABSTRACT. We study techniques for deciding the computational complexity of infinite-domain constraint satisfaction problems. For certain fundamental algebraic structures Δ , we prove definability dichotomy theorems of the following form: for every first-order expansion Γ of Δ , either Γ has a quantifier-free Horn definition in Δ , or there is an element d of Γ such that all non-empty relations in Γ contain a tuple of the form (d, \dots, d) , or all relations with a first-order definition in Δ have a primitive positive definition in Γ .

The results imply that several families of constraint satisfaction problems exhibit a complexity dichotomy: the problems are in P or NP-hard, depending on the choice of the allowed relations. As concrete examples, we investigate fundamental algebraic constraint satisfaction problems. The first class consists of all first-order expansions of $(\mathbb{Q}; +)$. The second class is the affine variant of the first class. In both cases, we obtain full dichotomies by utilising our general methods.

1. INTRODUCTION

Constraint satisfaction problems (CSPs) are computational problems that appear in almost every area of computer science such as artificial intelligence, graph algorithms, scheduling, combinatorics, and computer algebra. Depending on the type of constraints that are allowed in the input instances of a CSP, the computational complexity of a CSP is usually polynomial (we will call these CSPs tractable), or NP-hard. In the last decade, a lot of progress was made to find general criteria that imply that a CSP is tractable, or that it is NP-hard. Such results have been obtained for constraint languages over finite domains [9, 11, 12, 17], but also for constraint languages over infinite domains that are ω -categorical (for formal definition of these concepts see Section 2). For example, it has been shown that for every structure Γ with a first-order definition in $(\mathbb{Q}; <)$ the problem $\text{CSP}(\Gamma)$ is in P if it falls into one out of nine classes, and is NP-hard otherwise [6].

Lately, many researchers have been fascinated by a conjecture due to Feder and Vardi [15] which is known as the *dichotomy conjecture*. This conjecture says that every CSP with a finite domain constraint language is either tractable (i.e., in P) or NP-complete. According

to a well-known result by Ladner [19], there are NP-*intermediate* computational problems, i.e., problems in NP that are neither tractable nor NP-complete (unless $P=NP$). But the problems that are given in Ladner’s construction are extremely artificial. The question why there are so few candidates for natural NP-intermediate problems is one of the mysteries in complexity theory.

Any outcome of the dichotomy conjecture is probably surprising: a negative answer would finally provide relatively natural NP-intermediate problems, which would be of interest in complexity theory. A positive answer probably comes with a criterion which describes the NP-hard CSPs (and it would probably even provide algorithms for the tractable CSPs). But then we would have a rich catalogue of computational problems where the computational complexity is known. Such a catalogue would be a valuable tool for deciding the complexity of computational problems in the mentioned application areas: since CSPs are abundant, one might derive algorithmic results by reducing the problem of interest to a known tractable CSP, and one might derive hardness results by reducing a known NP-hard CSP to the problem of interest.

In this article, we study two natural classes of infinite domain constraint languages, and show that the corresponding CSPs do exhibit a complexity dichotomy. To the best of our knowledge, this is the first systematic complexity result for classes of structures that are not ω -categorical. The first class consists of all first-order expansions of $(\mathbb{Q}; \{(x, y, z) \mid x + y = z\})$ (i.e., we add relations to $(\mathbb{Q}; \{(x, y, z) \mid x + y = z\})$ that are first-order definable in $(\mathbb{Q}; \{(x, y, z) \mid x + y = z\})$). The second class is an affine version of the first class, and consists of all first-order expansions of $(\mathbb{Q}; \{(a, b, c, d) \mid a - b + c = d\})$. That the structures $(\mathbb{Q}; \{(x, y, z) \mid x + y = z\})$ and $(\mathbb{Q}; \{(a, b, c, d) \mid a - b + c = d\})$ are not ω -categorical follows immediately from the theorem by Engeler, Ryll-Nardzewski, and Svenonius (cf. Theorem 6.3.1 in [16]). It is even the case that the corresponding CSPs cannot be formulated by any ω -categorical template; the basic proof idea is presented in [1, Proposition 1]; also see [4].

Our results follow from theorems about primitive positive definability: we show that for every relation R with a first-order definition in $(\mathbb{Q}; +)$, either R has a quantifier-free Horn definition in $(\mathbb{Q}; +)$, or R contains the tuple $(0, \dots, 0)$, or all relations with a first-order definition in $(\mathbb{Q}; +)$ have a primitive positive definition in $(\mathbb{Q}; +, R)$. The analogous result also holds for the affine case. The techniques that we use to prove these two definability theorems are more general than the two classification results, and they are very different in nature. One technique applies for structures ‘that have little structure’; to be precise, for *all* structures Γ where $=$ and \neq are the only primitive positive definable non-trivial binary relations (Section 5). In particular, they apply to structures with a 2-transitive automorphism group. The other technique applies for structures ‘with a lot of structure’; informally, it applies whenever we can find a primitive positive definition for the line between two points in \mathbb{Q}^k (Section 4).

The rest of this paper is organised as follows: in Section 2, we provide some background material on constraint satisfaction and logic. A tractability result for templates that have a quantifier-free Horn definition in $(\mathbb{Q}; +)$ is presented in Section 3. The classification result for $(\mathbb{Q}; +)$ can be found in Section 4 while the results for the affine case are collected in Section 5. Finally, a number of open questions and directions for future work can be found in Section 6.

2. PRELIMINARIES

Let $\Gamma = (D; R_1, \dots, R_n)$ be a relational structure¹ with domain D (which will usually be infinite) and finitely many relations R_1, \dots, R_n . The *constraint satisfaction problem* for Γ (short, $\text{CSP}(\Gamma)$) is the computational problem to decide whether a given primitive positive sentence Φ involving relation symbols for the relations in Γ is true in Γ . A first-order formula is called *primitive positive* if it is of the form

$$\exists x_1, \dots, x_n. \psi_1 \wedge \dots \wedge \psi_m$$

where ψ_i are atomic formulas, i.e., formulas of the form $x = y$ or $R(x_{i_1}, \dots, x_{i_k})$ with R the relation symbol for a k -ary relation from Γ . We call such a formula a *pp-formula*. The conjuncts in a pp-formula Φ are also called the *constraints* of Φ . We also refer to Γ as a *constraint language* (it is also often called the *template*) of $\text{CSP}(\Gamma)$.

We say that a first-formula ϕ *defines* a relation R in Γ when $\phi(a_1, \dots, a_k)$ holds in Γ iff $(a_1, \dots, a_k) \in R$. If ϕ is primitive positive, we call R *primitive positive definable* (*pp-definable*) over Γ . The following simple but important result explains the importance of primitive positive definability for constraint satisfaction problems.

Lemma 2.1. Let Γ be a relational structure and Γ' be an expansion of this structure by a pp-definable relation R over Γ . Then $\text{CSP}(\Gamma)$ is polynomial-time equivalent to $\text{CSP}(\Gamma')$.

Lemma 2.1 will be used extensively in the sequel and we will not make explicit references to it. Another important class of formulas are *Horn* formulas; a first-order formula in conjunctive normal form is Horn if and only if each clause contains at most one positive literal. A relation R is called quantifier-free Horn definable over Γ if there exists a quantifier-free Horn formula that defines R in Γ . Note that Lemma 2.1 does not hold if we replace ‘pp-definable’ with ‘Horn definable’.

By choosing an appropriate structure Γ many computational problems that have been studied in the literature can be formulated as $\text{CSP}(\Gamma)$ (see e.g. [1, 10, 11]). It turns out very often that the structure Γ can be chosen to be ω -categorical. A structure is called *ω -categorical* if the set of all first-order sentences that is true in the structure has only one countable model, up to isomorphism. A famous example of an ω -categorical structure is $(\mathbb{Q}; <)$. The condition of ω -categoricity is interesting for constraint satisfaction, because the so-called *universal-algebraic approach*, which is currently intensively studied for finite constraint languages, applies—at least in principle—also for ω -categorical structures (see e.g. [6] for an application of the universal-algebraic approach to CSPs for constraint languages over infinite domains). In this article, we demonstrate that systematic complexity classification can be performed for constraint languages over infinite domains even if the constraint languages are not ω -categorical.

Example. Let Γ denote the structure

$$(\mathbb{Q}; \{(x, y, u, v) \mid (x = 2y \vee y = u + v) \wedge x \neq u\})$$

It can be shown that $\text{CSP}(\Gamma)$ *cannot* be formulated with an ω -categorical template (for a very similar proof, see [1]; a necessary and sufficient condition about which CSPs can be formulated with ω -categorical templates can be found in [4]). One can show that the relations $\{(x, y) \mid x \neq y\}$ and $\{(x, y, z) \mid x = y + z\}$ have pp-definitions in Γ . It is

¹Our terminology is standard; all notions that are not introduced in the article can be found in standard text books, e.g., in [16].

now straightforward to determine the computational complexity of $\text{CSP}(\Gamma)$ by combining Lemma 2.1 and our classification result (Corollary 4.6).

We will sometimes consider the automorphism group $\text{Aut}(\Gamma)$ of a template Γ over a domain D , i.e., the group formed by the set of all automorphisms² of Γ with respect to functional composition. An *orbit* of $\text{Aut}(\Gamma)$ on D^2 is a set of the form $\{(\alpha(a), \alpha(b)) \mid \alpha \in \text{Aut}(\Gamma)\}$, for some $a, b \in D$. We note that pairs from the same orbit satisfy the same first-order formulas.

Let D be an arbitrary infinite set and arbitrarily choose an element $d \in D$. The complexity of $\text{CSP}(\Gamma)$ where Γ has a first-order definition in $(D; =)$ (so-called *equality languages*) has been classified in [5]. We note that if R is first-order definable in $(D; =)$ and $(d, \dots, d) \in R$, then $(d', \dots, d') \in R$ for every $d' \in D$. Thus, the exact choice of d is irrelevant when stating the following theorem.

Theorem 2.2 (of [5]). Let Γ be a template with a first-order definition in $(D; =)$. Then, all relations in Γ have a quantifier-free Horn definition in $(D; =)$, or all non-empty relations in Γ contain the tuple (d, \dots, d) , or else every first-order definable relation in $(D; =)$ has a pp-definition in Γ . In the last case, $\text{CSP}(\Gamma)$ is NP-complete.

Instead of using Theorem 2.2 in its full generality, it will be sufficient to use a simple corollary. For any set D , the relation S_D denotes the relation

$$\{(x, y, z) \in D^3 \mid y \neq z \wedge (x = y \vee x = z)\}.$$

Corollary 2.3. Let D be an infinite set. Every first-order definable relation in $(D; =)$ has a pp-definition in $(D; S_D)$.

Proof. The relation S_D has a first-order definition in $(D; =)$ and does not contain the tuple (d, d, d) . It is easy to verify that S_D has no quantifier-free Horn definition in $(D; =)$ so every first-order definable relation in $(D; =)$ has a pp-definition in $(D; S_D)$ by Theorem 2.2. \square

3. TRACTABILITY

For all relational structures Γ with a quantifier-free Horn definition in $(\mathbb{Q}; +)$, the problem $\text{CSP}(\Gamma)$ can be solved in polynomial time. This follows from a more general algorithmic result in [18]. However, the algorithm presented there solves a linear number of linear programs, and thus the best known algorithms have a rather high worst-case running time. We present a more efficient algorithm for the special case that is relevant in our paper. We denote by $O^\sim(f(N))$ the class of all functions of asymptotic growth at most $f(N)$ up to poly-logarithmic factors.

Proposition 3.1. Let Γ be a relational structure whose relations have a quantifier-free Horn definition in $(\mathbb{Q}; +)$. Then there is an algorithm that solves $\text{CSP}(\Gamma)$ in time $O^\sim(N^4)$ where N is the size of the input.

The algorithm we present in the proof of Proposition 3.1 is a combination of general techniques in constraint satisfaction [2, 13] and a polynomial implementation of Gaussian elimination algorithm on rational data. Since the input of $\text{CSP}(\Gamma)$ consists of a primitive

²Isomorphisms between Γ and Γ .

positive sentence whose atomic formulas are of the form $R(x_1, \dots, x_k)$ where R is quantifier-free Horn definable over $(\mathbb{Q}; +)$, we can as well assume that the input to our problem consists of a set of Horn clauses over $(\mathbb{Q}; +)$.

We have to make some remarks about the worst-case running time of the Gaussian elimination algorithm. It is well-known that the Gaussian elimination requires $O(n^2m)$ many arithmetic operations on rational numbers, where m is the number of equations and n is the number of variables. In our algorithm, we have to solve a linear number of linear equation systems S_1, \dots, S_m ; however, system S_{i+1} is obtained from system S_i by adding a single linear equation. Since the Gaussian algorithm can be presented in such a way that it computes a system in triangular form, adding successively equation by equation, the overall costs for solving S_1, \dots, S_m equals the cost to solve S_m with Gaussian elimination.

Also recall that the size of the numbers involved when performing the Gaussian elimination algorithm might grow exponentially when implemented without care. However, when we use the Euclidean algorithm to shorten the coefficients during the elimination process, the Gaussian elimination algorithm can be shown to be polynomial [14]. We are only interested in deciding solvability of linear equation systems, and not constructing solutions, and so we even have *linear* bounds (in the input size) on the representation size of all numbers involved in deciding solvability for linear equation systems over the rational numbers with Gaussian elimination (see [22], proof of Theorem 3.3). Finally we remark that the most costly arithmetic operation that has to be performed on rational numbers during the elimination process is multiplication, and multiplication can be performed in time $O(s \log s \log \log s)$, where s denotes the representation size of the two rational numbers (in bits). Hence, the overall running time for solving S_1, \dots, S_m with the discussed implementation of the Gaussian elimination algorithm is in $O^\sim(N^4)$.

We will show that our algorithm for $\text{CSP}(\Gamma)$ can be implemented such that it has the same overall asymptotic worst-case complexity.

```

Solve( $\Phi$ )
// Input: An instance  $\Phi$  of  $\text{CSP}(\Gamma)$ 
// where all relations in  $\Gamma$  have a quantifier-free Horn definition in  $(\mathbb{Q}; +)$ 
// Output: yes if  $\Phi$  is true in  $\Gamma$ , false otherwise
Let  $\mathcal{C}$  be the set of all Horn-clauses from each constraint in  $\Phi$ 
Let  $\mathcal{U}$  be the subset of  $\mathcal{C}$  that only contains clauses with a single positive literal.
Do
  For all negative literals  $\neg\phi$  in clauses from  $\mathcal{C}$ 
    If  $\mathcal{U}$  implies  $\phi$  delete the negative literal  $\neg\phi$  from all clauses in  $\mathcal{C}$ .
  If  $\mathcal{C}$  contains an empty clause then return unsatisfiable.
  If  $\mathcal{C}$  contains a clause with a single positive literal  $\psi$ , add  $\{\psi\}$  to  $\mathcal{U}$ .
Loop until no literal has been deleted
Return satisfiable.

```

Figure 1: An algorithm for the constraint satisfaction problem where all constraint relations have a quantifier-free Horn definition in $(\mathbb{Q}; +)$.

Proof of Proposition 3.1. We first discuss the correctness of the algorithm shown in Figure 1, and then explain how to implement the algorithm such that it achieves the desired running time.

When \mathcal{U} logically implies ϕ then the negative literal $\neg\phi$ is never satisfied and can be deleted from all clauses without affecting the set of solutions. Since this is the only way how literals can be deleted from clauses, it is clear that if one clause becomes empty the instance is unsatisfiable.

If the algorithm terminates with *yes*, then no negation of a disequality is implied by \mathcal{U} . If r is the rank of the linear equation system defined by \mathcal{U} , we can use the Gaussian elimination algorithm as described above to eliminate from all literals in the remaining clauses r of the variables. Let S be the maximal sum of the absolute values of all coefficients in one of the remaining inequalities plus one. Then setting the i -th variable to S^i satisfies all clauses.

To see this, take any disequality, and assume that i is the highest variable index in this disequality. Order the disequality in such a way that the variable with highest index is on one side and all other on the other side of the \neq sign. The absolute value on the side with the i -th variable is at least S^i . The absolute value on the other side is less than $S^i - S$, since all variables have absolute value less than S^{i-1} and the sum of all coefficients is less than $S - 1$ in absolute value. Hence, both sides of the disequality have different absolute value, and the disequality is satisfied. Since all remaining clauses have at least one disequality, all constraints are satisfied.

We finally explain how to implement the algorithm such that it runs in time $O^\sim(N^4)$. To decide whether \mathcal{U} implies an equality ϕ , we compute in each iteration of the main loop the triangular normal form for the linear equation system determined by \mathcal{U} as described before the statement of the Proposition. The overall costs to do this are in $O^\sim(N^4)$. Moreover, for each negative literal we maintain an equation where we eliminate as many variables as possible using the computed triangular normal form. If one of these equations becomes trivial (i.e. is the form $a = a$) we conclude that the equation is implied by \mathcal{U} . The overall costs for doing this is also bounded by $O^\sim(N^4)$ by a very similar argument as given before the statement of the proposition. With appropriate straightforward data structures, the total costs for removing negated literals $\neg\phi$ from all clauses when ϕ is implied by \mathcal{U} is linearly bounded in the input size since each literal can be removed at most once. \square

4. THE RATIONAL NUMBERS WITH ADDITION

In this section we present the complexity classification for first-order expansions of $(\mathbb{Q}; \{(x, y, z) \mid x + y = z\})$. We begin in Section 4.1 with a result about the pp-definability of the disequality relation \neq in first-order expansions of $(\mathbb{Q}; \{(x, y, z) \mid x + y = z\})$. When the relation \neq is pp-definable, we show that also the relation $S_{\mathbb{Q}}$ (defined in Section 2 as the relation $\{(x, y, z) \in \mathbb{Q}^3 \mid x \neq z \wedge (x = y \vee y = z)\}$) is pp-definable whenever the constraint language contains a relation R that is first-order, but not quantifier-free Horn definable in $(\mathbb{Q}; +)$; this is shown in Section 4.2. Finally, Section 4.3 completes the classification for first-order expansions of $(\mathbb{Q}; \{(x, y, z) \mid x + y = z\})$.

4.1. Definability of Disequality.

Lemma 4.1. For any structure Γ with a first-order definition in $(\mathbb{Q}; +)$, the first-order definable relations in Γ are a subset of $\{\mathbb{Q}, \mathbb{Q} \setminus \{0\}, \{0\}, \emptyset\}$.

Proof. Let R be a unary relation with a first-order definition in $(\mathbb{Q}; +)$. The statement is clear if R does not contain any element distinct from 0, so let a be from $\mathbb{Q} \setminus \{0\}$. We have to show that $R = \mathbb{Q}$ or $R = \mathbb{Q} \setminus \{0\}$. Observe that for any $c \in \mathbb{Q}$, $c \neq 0$, the mapping $x \mapsto cx$

is an automorphism of Γ . Hence, for any $b \neq 0$ there is an automorphism of $(\mathbb{Q}; +)$ that maps a to b . Since automorphisms preserve first-order formulas, so $b \in R$ and the claim follows. \square

Note that $x = 0$ is equivalent to $x + x = x$ and hence the relation $\{0\}$ is pp-definable over $(\mathbb{Q}; +)$; thus we can use 0 freely as a constant symbol in pp-definitions over Γ .

Proposition 4.2. Let Γ be a first-order expansion of $(\mathbb{Q}; +)$ containing a non-empty relation R such that $R(x, \dots, x)$ is false for any x . Then \neq is pp-definable in Γ .

Proof. Observe that if the set $\mathbb{Q} \setminus \{0\}$ has a pp-definition $\phi(u)$ in Γ , then the pp-formula

$$\exists u, y'. \phi(u) \wedge y + y' = 0 \wedge x + y' = u$$

defines $x \neq y$ over Γ .

Let S be a non-empty pp-definable relation in Γ of minimal arity such that $S(x, \dots, x)$ defines the empty set. Let k be the arity of S . First, assume that $S(x_1, x_2, \dots, x_k) \wedge x_1 = x_2$ is satisfiable. Then the $(k-1)$ -ary relation $S'(x_2, \dots, x_k)$ defined by $S(x_2, x_2, \dots, x_k)$ is non-empty, and $S'(x, \dots, x)$ defines the empty set; this is in contradiction to the choice of S .

Assume next that $S(x_1, \dots, x_k) \wedge x_1 = x_2$ is unsatisfiable. Define the unary relation $T(x)$ by

$$\exists x_3, \dots, x_k. S(x, 0, x_3, \dots, x_k)$$

and the unary relation $U(y)$ by

$$\exists x_1, x_3, \dots, x_k. S(x_1, y, x_3, \dots, x_k).$$

By Lemma 4.1, both T and U are from $\{\mathbb{Q}, \mathbb{Q} \setminus \{0\}, \{0\}, \emptyset\}$. The relation T cannot be equal to $\{0\}$ or to \mathbb{Q} since this contradicts the assumption that $S(x_1, x_2, \dots, x_k) \wedge x_1 = x_2$ is unsatisfiable. If T is equal to $\mathbb{Q} \setminus \{0\}$, then by the initial observation \neq is pp-definable in Γ and we are done. We conclude that $T = \emptyset$ and hence $0 \notin U$. Since U is non-empty, it must be the case that $U = \mathbb{Q} \setminus \{0\}$, and again by the initial observation \neq is pp-definable in Γ . \square

4.2. Definability of $S_{\mathbb{Q}}$. The rational numbers with addition (and also the real numbers with addition) admit *quantifier elimination*, i.e., every relation with a first-order definition in $(\mathbb{Q}; +)$ also has a quantifier-free definition over $(\mathbb{Q}; +)$. This follows from the more general fact that the first-order theory of torsion-free divisible abelian groups admits quantifier elimination (see e.g. Theorem 3.1.9 in [20]).

The first lemma allows us to freely use certain expressions in pp-definitions over $(\mathbb{Q}; +)$.

Lemma 4.3. The relation $\{(x_1, \dots, x_l) \mid r_1 x_1 + \dots + r_l x_l = 0\}$ is pp-definable in $(\mathbb{Q}; +)$ for arbitrary $r_1, \dots, r_l \in \mathbb{Q}$.

Proof. First observe that we can assume that r_1, \dots, r_l are integers, because we can multiply the equation $r_1 x_1 + \dots + r_l x_l = 0$ by the least common multiple of the denominators of r_1, \dots, r_l and obtain an equivalent equation. The proof is by induction on l . We first consider that case that $l = 1$. If $r_1 = 0$, there is nothing to show. Otherwise, the formula $r_1 x_1 = 0$ is equivalent to $x_1 + x_1 = x_1$. Hence, we can in particular use expressions of the form $x = 0$ and $x + y = 0$ in pp-definitions over $(\mathbb{Q}; +)$ with variables x, y . If $l = 2$, and

$r_1 = 0$ or $r_2 = 0$, then we can argue as in the case $l = 1$. If r_1 and r_2 are both positive or both negative, then $r_1x_1 + r_2x_2 = 0$ is equivalent to

$$\begin{aligned} \exists u_1, \dots, u_{r_1}, v_1, \dots, v_{r_2}. u_1 = x_1 \wedge v_1 = x_2 \wedge u_{r_1} + v_{r_2} = 0 \wedge \\ \bigwedge_{i=1}^{r_1-1} x_1 + u_i = u_{i+1} \wedge \bigwedge_{i=1}^{r_2-1} x_2 + v_i = v_{i+1}. \end{aligned}$$

If r_1 and r_2 have different signs, we replace the conjunct $u_{r_1} + v_{r_2} = 0$ in the formula above by $u_{r_1} = v_{r_2}$.

Now suppose that $l > 2$. By the inductive assumption, there is a pp-definition ϕ_1 for $r_1x_1 + r_2x_2 + u = 0$ and a pp-definition ϕ_2 for $r_3x_3 + \dots + r_lx_l + v = 0$. Then $\exists u, v. \phi_1 \wedge \phi_2 \wedge u + v = 0$ is a pp-definition for $r_1x_1 + \dots + r_lx_l = 0$. \square

In the following, R denotes a relation with a quantifier-free first-order definition ϕ in $(\mathbb{Q}; +)$. A quantifier-free first-order formula ϕ in conjunctive normal form is called *reduced* if every formula obtained from ϕ by removing a literal is not equivalent to ϕ (this concept was introduced in [3]). Clearly, such a reduced definition of R always exists, because we can find one by successively removing literals from ϕ . Note that if l is a literal from ϕ , then $\neg l$ can be written as a pp-formula over a structure that contains \neq and $+$.

Lemma 4.4. If R is first-order, but not quantifier-free Horn definable in $(\mathbb{Q}; +)$, then $S_{\mathbb{Q}}$ has a pp-definition in $(\mathbb{Q}; R, +, \neq)$.

Proof. Let $T(x, y) \subsetneq \mathbb{Q}^2$ be the binary relation defined by $x \neq 0 \wedge (y = 0 \vee x = y)$. We first prove that T has a pp-definition in $(\mathbb{Q}; R, +, \neq)$. Let ϕ be a reduced first-order definition of R , and let C be a clause of ϕ with two positive literals l_1 and l_2 . Because ϕ is reduced, there are $p, q \in R$ such that p satisfies l_1 and does not satisfy all other literals in C , and q satisfies l_2 but does not satisfy all other literals in C .

We claim that the following pp-formula is logically equivalent to $x \neq 0 \wedge (y = 0 \vee x = y)$.

$$\begin{aligned} \exists z_1, \dots, z_k. \quad x \neq 0 \quad \wedge \quad \bigwedge_{i=1}^k z_i = p_i x + (q_i - p_i) y \quad \wedge \\ \bigwedge_{l \in C \setminus \{l_1, l_2\}} \neg l \quad \wedge \quad R(z_1, \dots, z_k) \end{aligned}$$

Let $x \neq 0$ be arbitrary. Suppose that $y = 0$. Then the assignment $z_1 = p_1x, \dots, z_k = p_kx$ obviously satisfies the first line in the pp-formula. Recall that $p \in R$ and p does not satisfy all literals in C except for l_1 . The function $f(a) = x \cdot a$ is in $\text{Aut}(\mathbb{Q}; +)$ whenever $x \neq 0$. Consequently, $f \in \text{Aut}(\mathbb{Q}; R)$, too, and the second line in the formula is satisfied as well. Now suppose that $x = y$. Then the assignment $z_1 = q_1x, \dots, z_k = q_kx$ obviously satisfies the first line in the pp-formula. By construction, $q \in R$ and q does not satisfy all literals in C except for l_2 . Again we conclude that the second line in the formula is also satisfied.

For the opposite direction, suppose that $x, y \in \mathbb{Q}$ satisfy the pp-formula. Because of the first line of the formula, $x \neq 0$. Let z_1, \dots, z_k be the k elements whose existence is asserted in the first line of the formula. Note that the equations of the first line imply that (z_1, \dots, z_k) lies on the line $L \subset \mathbb{Q}^k$ defined by px and qx . Because the formula contains the conjunct $R(z_1, \dots, z_k)$, the clause C in ϕ is satisfied by z_1, \dots, z_k . Since z_1, \dots, z_k also satisfies the conjunction of all negated literals in C except for the positive literals l_1 and l_2 , at least one of these two literals l_1 and l_2 must be satisfied by z_1, \dots, z_k .

Suppose first that l_1 is satisfied. The line L does not lie completely within the subspace of \mathbb{Q}^k defined by l_1 (because q does not satisfy l_1 , and neither does qx). Hence, L intersects this subspace in at most one point. Because p and hence also $px \in L$ satisfies l_1 , we have thus shown that (z_1, \dots, z_k) equals px . Since $p \neq q$ we conclude that $y = 0$ by the equations in the second line of the formula. Now, consider the case that l_2 is satisfied. Similarly as in the last case, L intersects the subspace defined by l_2 in at most one point. Because $q \in L$ satisfies l_2 , we have shown that (z_1, \dots, z_k) equals q . The equations in the second line of the formula then imply that $x = y$.

Finally, we prove that $S_{\mathbb{Q}}(u, v, w)$ has the following pp-definition in $(\mathbb{Q}; +, T)$:

$$\exists x, y. x + v = w \wedge y + v = u \wedge T(x, y).$$

Suppose first that $(u, v, w) \in S_{\mathbb{Q}}$. Note that $x = w - v$ is not equal to 0 because $v \neq w$. If $u = v$, then $y = 0$, and if $u = w$, then $x = w - v = u - v = y$ so $T(x, y)$ is satisfied.

Conversely, suppose that $(x, y) \in \mathbb{Q}^2$ satisfies the pp-formula above. The formula $T(x, y)$ implies that $x \neq 0$ and hence $w \neq v$. Moreover, $T(x, y)$ implies that $y = 0$ or $x = y$. If $y = 0$, then $u = v$ and $(u, v, w) \in S_{\mathbb{Q}}$. If $x = y$, then $w - v = u - v$ and hence $u = w$. Again (u, v, w) is in $S_{\mathbb{Q}}$. \square

4.3. Classification Result. We will now use Lemma 4.4 in order to prove the following definability result.

Theorem 4.5. Let Γ be first-order expansion of $(\mathbb{Q}; +)$. Then, either

- each relation in Γ has a quantifier-free Horn definition in $(\mathbb{Q}; +)$, or
- every non-empty relation of Γ contains a tuple of the form $(0, \dots, 0)$, or
- every first-order definable relation in $(\mathbb{Q}; +)$ has a pp-definition in Γ .

Proof. Suppose that there is a non-empty k -ary relation R of Γ that does not contain the tuple $(0, \dots, 0)$. Then the $(k+1)$ -ary relation $R'(x_1, \dots, x_{k+1})$ defined by $R(x_1, \dots, x_k) \wedge x_{k+1} = 0$ is non-empty, and the relation defined by $R'(x, \dots, x)$ is empty. So we can apply Proposition 4.2 and find that \neq is pp-definable in $(\mathbb{Q}; +, R')$ and hence also in Γ . So assume in the following without loss of generality that Γ contains the relation \neq .

Suppose that one of the relations of Γ does not have a quantifier-free Horn definition in $(\mathbb{Q}; +)$. Lemma 4.4 implies that the relation $S_{\mathbb{Q}}$ has a pp-definition in Γ , and Corollary 2.3 implies that every relation with a first-order definition in $(\mathbb{Q}; =)$ has a pp-definition in Γ .

Let R be a relation with a first-order definition ϕ in $(\mathbb{Q}; +)$. To find a pp-definition for R in Γ , we introduce a variable u for every atomic formula of the form $x + y = z$ in ϕ . For each atomic formula ψ in ϕ of the form $x + y = z$, we replace ψ by $u_{\psi} = z$ for a new variable u_{ψ} . The resulting formula consists of a boolean combination of atomic formulas of the form $x = y$, which we know has a pp-definition ϕ' in Γ . For each atomic formula ψ in ϕ we add the conjunct $x + y = u_{\psi}$ to ϕ' , and finally existentially quantify over all new variables. It is straightforward to verify that the resulting formula is a pp-definition of R in Γ . \square

Theorem 4.5 has immediate consequences for the computational complexity of constraint satisfaction.

Corollary 4.6. Let Γ be a structure with a finite relational signature and a first-order definition in $(\mathbb{Q}; +)$ that contains the relation $\{(x, y, z) \mid x + y = z\}$. Then $\text{CSP}(\Gamma)$ is in P if all relations in Γ have a quantifier-free Horn definition over $(\mathbb{Q}; +)$, or if all non-empty relations contain a tuple of the form $(0, \dots, 0)$, and is NP-hard otherwise.

Proof. If all relations in Γ have a quantifier-free Horn definition over $(\mathbb{Q}; +)$, then Proposition 3.1 implies that $\text{CSP}(\Gamma)$ is in P. Otherwise, Theorem 4.5 implies that in particular the relation defined by $(x = y \wedge y \neq z) \vee (x \neq y \wedge y \neq z)$ is pp-definable in Γ . It follows from Theorem 2.2 that the constraint satisfaction problem for this ternary relation is NP-hard. \square

5. AFFINE STRUCTURES OVER THE RATIONAL NUMBERS

We will now consider affine additive structures over \mathbb{Q} . The structure of this section is very similar to the structure of Section 4: we begin by studying the definability of \neq (Section 5.1) and of S_D (in Section 5.2) and use these results to completely classify the problem in Section 5.3. The main proof in Section 5.2, however, is very different from the corresponding proof in Section 4.2.

Let us now formally define the problem at hand: define the operation $f : \mathbb{Q}^3 \rightarrow \mathbb{Q}$ by $f(a, b, c) = a - b + c$. We study the constraint satisfaction problem for templates Γ with a first-order definition in $(\mathbb{Q}; f)$ that contain the relation $\{(a, b, c, d) \mid a - b + c = d\}$.

5.1. Definability of Disequality.

Lemma 5.1. Let Γ be a structure with a first-order definition in $(\mathbb{Q}; f)$. Then there are at most four first-order definable binary relations: the empty relation, the full relation, the relation \neq , and the relation $=$.

Proof. It suffices to show that $\text{Aut}(\Gamma)$ has precisely two orbits on \mathbb{R}^2 , namely

$$O_1 = \{(x, x) \mid x \in \mathbb{Q}\} \quad \text{and} \quad O_2 = \{(x, y) \mid x, y \in \mathbb{Q}, x \neq y\}.$$

These two orbits clearly partition \mathbb{Q}^2 . It is obvious that O_1 is an orbit, because for every $c \in \mathbb{Q}$ the mapping $x \mapsto x + c$ is an automorphism of $(\mathbb{Q}; f)$ and hence of Γ . To see that O_2 is an orbit of pairs of reals, we apply linear interpolation: let $(a, b) \in O_2$ and $(c, d) \in O_2$ be arbitrary. The mapping $x \mapsto \frac{c-d}{a-b}(x - a) + c$ maps (a, b) to (c, d) and it is an automorphism of $(\mathbb{Q}; f)$, and hence of Γ . \square

In the proof of Lemma 5.1 we have in fact verified that the automorphism group of Γ is *2-transitive*, i.e., that there is only one orbit of pairs of distinct elements with respect to the componentwise action of the automorphism group of Γ on pairs.

Theorem 5.2 (from [6]). Let Γ be a relational structure with a 2-transitive automorphism group. If there is no pp-definition of \neq , then there is an element x of Γ such that every non-empty relation in Γ contains a tuple of the form (x, \dots, x) .

5.2. Definability of S_D . The central step of the classification is the following result concerning pp-definability.

Lemma 5.3. Let Γ be a relational structure over an infinite domain D such that D^2 , $=$, \neq , and \emptyset are the only pp-definable binary relations. Suppose that Γ contains a relation Q such that there are pairwise distinct $1 \leq i, j, k, l \leq n$ for which the following conditions hold:

- (1) $Q(x_1, \dots, x_n) \wedge x_i \neq x_j$ is satisfiable;
- (2) $Q(x_1, \dots, x_n) \wedge x_k \neq x_l$ is satisfiable;
- (3) $Q(x_1, \dots, x_n) \wedge x_i \neq x_j \wedge x_k \neq x_l$ is unsatisfiable.

Then S_D has a pp-definition in Γ .

We simplify the proof of Lemma 5.3 by first proving a slightly restricted version:

Lemma 5.4. Let Γ be a relational structure over an infinite domain D such that D^2 , $=$, \neq , and \emptyset are the only pp-definable binary relations. Suppose that Γ contains a relation Q such that there are $1 \leq i, j, k \leq n$ for which the following conditions hold:

- (1) $Q(x_1, \dots, x_n) \wedge x_i \neq x_j$ is satisfiable;
- (2) $Q(x_1, \dots, x_n) \wedge x_i \neq x_k$ is satisfiable;
- (3) $Q(x_1, \dots, x_n) \wedge x_i \neq x_j \wedge x_i \neq x_k$ is unsatisfiable.

Then S_D has a pp-definition in Γ .

Proof. The indices i, j, k must be pairwise distinct, so suppose for the sake of notation that $i = 1, j = 2, k = 3$. Consider the relation R defined by

$$R(x_1, x_2, x_3) \equiv \exists x_4, \dots, x_n. Q(x_1, \dots, x_n) \wedge x_2 \neq x_3.$$

We first note that R is a non-empty relation: $Q(x_1, \dots, x_n)$ is satisfiable so the only way of making R empty is that every tuple (s_1, \dots, s_n) in Q satisfies $s_2 = s_3$. This is impossible since we know that there exists a tuple $(s_1, \dots, s_n) \in Q$ such that $s_1 \neq s_2$. This implies $s_1 \neq s_3$ and contradicts the third condition.

Arbitrarily choose a domain element a . We first show that there always exist elements $y, z \in D$ such that $(a, y, z) \in R$. Let $A = \{a \in D \mid \exists y, z. R(a, y, z)\}$ and note that A is pp-definable. We know that A is non-empty since R is non-empty. Now assume that $A \subsetneq D$. First suppose that $|A| = 1$. Then

$$A'(x, y) \equiv A(x) \wedge A(y)$$

is non-empty and a strict subset of the equality relation, a contradiction.

If $|A| > 1$, then consider the pp-definable relation

$$A'(x, y) \equiv A(x) \wedge A(y) \wedge x \neq y.$$

We see that $\emptyset \subsetneq A' \subsetneq \{(u, v) \in D^2 \mid u \neq v\}$ which contradicts the fact that the only non-trivial binary relations that are pp-definable from Γ are $=$ and \neq . Hence, $A = D$.

We now continue by considering the tuple $(a, y, z) \in R$. By the third condition, we see that at least one of y, z must equal a in order to satisfy R . Let us consider the case $R(a, a, z)$. Note that $(a, a, a) \notin R$ due to the literal $y \neq z$. We now show that $R(a, a, z)$ is satisfied by any choice of z except a . To see this, assume to the contrary that there is a domain element $b \neq a$ such that $(a, a, b) \notin R$. Define $R'(x, z) \equiv R(x, x, z)$ and note that $\emptyset \subsetneq R' \subsetneq \{(u, v) \in D^2 \mid u \neq v\}$ which contradicts the assumption that $=$ and \neq are the only non-trivial pp-definable binary relations. Similarly, one can show that $R(a, y, a)$ holds for all $y \neq a$. Therefore $R = S_D$. \square

Proof of Lemma 5.3. Assume for notational simplicity that $i = 1, j = 2, k = 3$, and $l = 4$. Define the 4-ary relation R by

$$R(x_1, x_2, x_3, x_4) \equiv \exists x_5, \dots, x_n. Q(x_1, \dots, x_n)$$

and consider the formula $\phi = R(x, y, x', y') \wedge R(z', y', z, y) \wedge x' \neq z'$. We claim that $\phi \wedge x \neq y$ and $\phi \wedge y \neq z$ are satisfiable while $\phi \wedge x \neq y \wedge y \neq z$ is not satisfiable. Then we can apply Lemma 5.4 and are done. First we make an observation:

Observation 1. Define relation R_1 such that

$$R_1(u, v) \equiv \exists x, y. R(x, y, u, v) \wedge x \neq y.$$

We know that $R(x, y, u, v) \wedge x \neq y$ is satisfiable so R_1 is a non-empty relation. Since $R_1(u, v) \wedge u \neq v$ is not satisfiable, we conclude that R_1 is a non-empty subset of the equality relation. Consequently, R_1 is the equality relation. Analogously, define R_2 such that

$$R_2(u, v) \equiv \exists z, y. R(u, v, z, y) \wedge z \neq y$$

and note that R_2 is the equality relation, too.

We now prove that $\phi \wedge x \neq y \wedge y \neq z$ is not satisfiable. By using Observation 1, it follows that any solution s satisfies $x' = y'$ and $y' = z'$ — this is impossible due to the clause $x' \neq z'$.

Next, we prove that $\phi \wedge x \neq y$ is satisfiable; the case $\phi \wedge y \neq z$ is symmetric. Consider the relation

$$U(u, v) \equiv \exists w. R(w, u, v, v) \wedge w \neq u.$$

By the conditions on R , we know that U is non-empty. Since U is binary, we also know that U either is the equality relation, the disequality relation, or the full relation. We conclude that U is non-empty and symmetric.

By Observation 1, the clause $x \neq y$ has the effect that every solution s must satisfy $x' = y'$. The solution also has to satisfy $x' \neq z'$ which implies that $y' \neq z'$. Observation 1 now tells us that $z = y$ and we conclude that every solution satisfies $x' = y'$ and $z = y$. We define

$$\phi' = R(x, y, x', x') \wedge R(z', y', z, z) \wedge x' \neq z' \wedge x \neq y$$

Thus, ϕ' is satisfiable if and only if $\phi \wedge x \neq y$ is satisfiable. We will now construct a concrete satisfying assignment s to the variables of ϕ' .

Arbitrarily choose a tuple $(a, b) \in U$ and let $s(y) = a$, $s(x') = b$. By the conditions on U , there exists an element c such that $(c, a, b, b) \in R$ and $c \neq a$; we let $s(x) = c$. Furthermore, we know that $s(x') = s(y')$ and $s(z) = s(y)$ so $s(y') = b$ and $s(z) = a$. At this point, we see that the assignment s satisfies the clauses $R(x, y, x', x')$ and $x \neq y$.

We know that $(a, b) \in U$ so $(b, a) \in U$, too, and there exists a value d such that $(d, b, a, a) \in R$ and $d \neq b$. Now, let $s(z') = d$ and note that $R(z', y', z, z)$ is satisfied by s . Finally, $s(x') = b \neq d = s(z')$ so the clause $x' \neq z'$ is satisfied and the proof is completed. \square

5.3. Classification Result. We are now ready to prove the classification result for the affine case.

Theorem 5.5. Let Γ be a first-order expansion of $(\mathbb{Q}; f)$. Then, either

- each relation in Γ has a quantifier-free Horn definition in $(\mathbb{Q}; f)$, or
- every non-empty relation of Γ contains a tuple of the form $(0, \dots, 0)$, or
- every first-order definable relation in $(\mathbb{Q}; f)$ has a pp-definition in Γ .

Proof. Suppose that there is a non-empty k -ary relation R of Γ that does not contain the tuple $(0, \dots, 0)$. The proof of Lemma 5.1 shows that Γ is 2-transitive, and hence by the contraposition of Theorem 5.2 the relation \neq is pp-definable. So assume in the following without loss of generality that Γ contains the relation \neq .

Let R be a relation in Γ that does not have a quantifier-free Horn definition in $(\mathbb{Q}; f)$. Let $\phi(x_1, \dots, x_n)$ be a reduced definition of R in $(\mathbb{Q}; f)$ (see Section 4). Then there must be a

clause C in ϕ with at least two positive literals $f(x_{i_1}, x_{i_2}, x_{i_3}) = x_{i_4}$ and $f(x_{j_1}, x_{j_2}, x_{j_3}) = x_{j_4}$. Let $Q(x_1, \dots, x_n, x_{n+1}, x_{n+2})$ be the relation defined by

$$\begin{aligned} & \phi(x_1, \dots, x_n) \wedge \bigwedge_{l \in C \setminus \{l_1, l_2\}} \neg l \wedge \\ & x_{n+1} = f(x_{i_1}, x_{i_2}, x_{i_3}) \wedge x_{n+2} = f(x_{j_1}, x_{j_2}, x_{j_3}). \end{aligned}$$

This relation Q is clearly pp-definable over $(\mathbb{Q}; R, f, \neq)$. We claim that Q satisfies the conditions of Lemma 5.3 (which is applicable due to Lemma 5.1) with respect to the arguments indexed by $i_4, n+1, j_4$, and $n+2$ (or the conditions of Lemma 5.4 if $i_4 = j_4$; this remark also applies to all other places where we appeal to Lemma 5.3). Since ϕ is reduced, there is a tuple $t \in R$ that satisfies l_2 and does not satisfy all other literals in ϕ . Now, the extended tuple $t_1 = (t[1], \dots, t[n], t[i_4], t[j_4])$ clearly satisfies Q , and we have $t_1[i_4] \neq t_1[n+2]$ as required in the conditions for Lemma 5.4. There is also a tuple $t_2 \in R$ that satisfies l_1 and does not satisfy all other literals in C , and we can argue similarly to find a second tuple showing the second condition of Lemma 5.3.

Finally, suppose for contradiction that there is a tuple t_3 in Q where $t_3[i_4] \neq t_3[n+1]$ and $t_3[j_4] \neq t_3[n+2]$. Because this tuple satisfies in particular the clause C from ϕ , the conjunct $\bigwedge_{l \in C \setminus \{l_1, l_2\}} \neg l$ implies that either l_1 or l_2 is satisfied. But then the equalities $x_{n+1} = f(x_{i_1}, x_{i_2}, x_{i_3})$ and $x_{n+2} = f(x_{j_1}, x_{j_2}, x_{j_3})$ imply that $t_3[i_4] = t_3[n+1]$ or $t_3[j_4] = t_3[n+2]$, a contradiction. Hence, Lemma 5.4 applies, $S_{\mathbb{R}}$ is pp-definable over $(\mathbb{Q}; Q)$ and therefore also over $(\mathbb{Q}; R, f, \neq)$ and Γ . The result follows from Corollary 2.3. \square

The next corollary is a direct consequence of Proposition 3.1, Theorem 5.5, and Corollary 2.3.

Corollary 5.6. Let Γ be an expansion of $(\mathbb{Q}; \{(a, b, c, d) \mid a - b + c = d\})$ by finitely many first-order definable relations. If each relation in Γ has a quantifier-free Horn definition in (\mathbb{R}, f) , or if each non-empty relation contains a tuple of the form $(0, \dots, 0)$, then $\text{CSP}(\Gamma)$ is in P. Otherwise, $\text{CSP}(\Gamma)$ is NP-hard.

6. CONCLUDING REMARKS

We have presented classification results for certain algebraic constraint satisfaction problems, and the results are to a large extent based on dichotomy results for logical definability. We feel that the results and ideas presented in this paper can be extended in many different directions. Hence, it seems worthwhile to provide some concrete suggestions for future work.

The results and proof techniques in Section 4 appear to be generalisable to many different templates defined over various structures. One example is the natural and important class of structures that are definable in *Presburger arithmetics* [21], i.e., structures that are first-order definable over the integers with addition $(\mathbb{Z}; +)$. We note that the following can be obtained by slightly modifying Corollary 4.6.

Corollary 6.1. Let Γ be a relational structure with a quantifier-free first-order definition in $(\mathbb{Z}; +)$ that contains the relation $\{(x, y, z) \mid x + y = z\}$. Then $\text{CSP}(\Gamma)$ is in P if all relations in Γ have a quantifier-free Horn definition over $(\mathbb{Z}; +)$, or if all non-empty relations contain a tuple of the form $(0, \dots, 0)$. Otherwise, $\text{CSP}(\Gamma)$ is NP-hard.

There is an important difference between this result and a full classification result: we have replaced *first-order definability* with *quantifier-free first-order definability* in the statement of the result, and the reason is that $(\mathbb{Z}; +)$ does not admit quantifier elimination. Is there still a complexity dichotomy if we look at the class of CSPs with an template that is first-order definable in $(\mathbb{Z}; +)$? This appears to be a difficult question.

The results presented in Section 5 have strong connections with earlier work on the complexity of *disjunctive* constraints [8, 13]. We say that \neq is *1-independent* with respect to a τ -structure Γ if and only if for every primitive positive τ -formula ϕ with free variables x, y, z, w the following holds: if $\phi \wedge x \neq y$ and $\phi \wedge z \neq w$ are satisfiable, then so is $\phi \wedge x \neq y \wedge z \neq w$. Assume that $\text{CSP}(\Gamma)$ is tractable and let Γ' denote the set of all relations that can be defined by (quantifier-free) conjunctions of disjunctions over Γ containing at most one literal that is not of the form $x \neq y$. The following has been shown in [8, 13]; it does not imply our result since it only makes a statement about a constraint language Γ' of the form described above.

Theorem 6.2 (from [8, 13]). Let Γ and Γ' be defined as above, and assume that $P \neq NP$. Then $\text{CSP}(\Gamma')$ is tractable if and only if \neq is 1-independent with respect to Γ .

We have already mentioned that the structures studied in this paper are in general not ω -categorical. However, torsion-free divisible abelian groups such as $(\mathbb{Q}; +)$ and all structures first-order definable in such groups are strongly minimal (see e.g. Corollary 3.1.11 in [20]), and hence categorical in all uncountable cardinals. This is interesting from a constraint satisfaction point of view because of the following preservation theorem.

Theorem 6.3 (of [4]). Let Γ be an uncountably categorical structure with a countable relational signature and an uncountable domain. Then a first-order definable relation R has a pp-definition in Γ if and only if R is preserved by all *infinitary* polymorphisms of Γ .

Note that this theorem is weaker than the corresponding theorem for ω -categorical structures [7], because we have to assume that R is first-order definable, and that R is not only preserved by the finitary, but also by the infinitary polymorphisms of Γ . Since our classification result is purely in terms of primitive positive definability of first-order definable relations, it is an interesting question to describe the polymorphisms that guarantee tractability for structures Γ with a first-order definition in $(\mathbb{Q}; +)$ (Theorem 6.3 shows that such polymorphisms do exist).

ACKNOWLEDGEMENTS

We want to thank Barnaby Martin for comments on an earlier version of the paper. Peter Jonsson is partially supported by the *Center for Industrial Information Technology* (CENIT) under grant 04.01 and by the *Swedish Research Council* (VR) under grant 2006-4532.

REFERENCES

- [1] M. Bodirsky. Constraint satisfaction problems with infinite templates. In H. Vollmer, editor, *Complexity of Constraints (a collection of survey articles)*, pages 196–228. Springer, LNCS 5250, 2008.
- [2] M. Bodirsky, H. Chen, J. Kara, and T. von Oertzen. Maximal infinite-valued constraint languages. *Theoretical Computer Science (TCS)*, 410:1684–1693, 2009. A preliminary version appeared at ICALP’07.

- [3] M. Bodirsky, H. Chen, and M. Pinsker. The reducts of equality up to primitive positive interdefinability. *To appear in the Journal of Symbolic Logic*, 2010.
- [4] M. Bodirsky, M. Hils, and B. Martin. On the scope of the universal-algebraic approach to constraint satisfaction. In *Proceedings of LICS'10*, 2010.
- [5] M. Bodirsky and J. Kára. The complexity of equality constraint languages. *Theory of Computing Systems*, 3(2):136–158, 2008. A conference version appeared in the proceedings of CSR'06.
- [6] M. Bodirsky and J. Kára. The complexity of temporal constraint satisfaction problems. In *Proceedings of STOC'08*, pages 29–38, 2008.
- [7] M. Bodirsky and J. Nešetřil. Constraint satisfaction with countable homogeneous templates. *Journal of Logic and Computation*, 16(3):359–373, 2006.
- [8] M. Broxvall, P. Jonsson, and J. Renz. Disjunctions, independence, refinements. *Artificial Intelligence*, 140(1/2):153–173, 2002.
- [9] A. Bulatov. A graph of a relational structure and constraint satisfaction problems. In *Proceedings of LICS'04, Turku, Finland*, 2004.
- [10] A. Bulatov, P. Jeavons, and A. Krokhin. The complexity of constraint satisfaction: An algebraic approach (a survey paper). In: *Structural Theory of Automata, Semigroups and Universal Algebra (Montreal, 2003)*, *NATO Science Series II: Mathematics, Physics, Chemistry*, 207:181–213, 2005.
- [11] A. Bulatov, A. Krokhin, and P. G. Jeavons. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34:720–742, 2005.
- [12] A. A. Bulatov and V. Dalmau. A simple algorithm for Mal'tsev constraints. *SIAM J. Comput.*, 36(1):16–27, 2006.
- [13] D. Cohen, P. Jeavons, P. Jonsson, and M. Koubarakis. Building tractable disjunctive constraints. *Journal of the ACM*, 47(5):826–853, 2000.
- [14] J. Edmonds. System of distinct representatives and linear algebra. *Journal of Research of the National Bureau of Standards*, 718(4):242–245, 1967.
- [15] T. Feder and M. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal on Computing*, 28:57–104, 1999.
- [16] W. Hodges. *A shorter model theory*. Cambridge University Press, Cambridge, 1997.
- [17] P. M. Idziak, P. Markovic, R. McKenzie, M. Valeriote, and R. Willard. Tractability and learnability arising from algebras with few subpowers. In *Proceedings of LICS'07*, pages 213–224, 2007.
- [18] P. Jonsson and C. Bäckström. A unifying approach to temporal constraint reasoning. *Artif. Intell.*, 102(1):143–155, 1998.
- [19] R. E. Ladner. On the structure of polynomial time reducibility. *JACM*, 22(1):155–171, 1975.
- [20] D. Marker. *Model Theory: An Introduction*. Springer, New York, 2002.
- [21] M. Presburger. über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. *Comptes Rendus du I congrès de Mathématiciens des Pays Slaves*, pages 92–101, 1929.
- [22] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley - Interscience Series in Discrete Mathematics and Optimization, 1998.